

**In The Drawings:**

The Examiner requested new corrected drawings because shading makes text unreadable. Applicant has corrected Figures 1 through 125 to remove shading as per the Examiner's request. 105 drawing pages including corrected Figures 1 through 125 are attached herewith.

## **REMARKS**

Claims 1-92 remain pending in the application. Claims 18-92 have been withdrawn. Reconsideration is respectfully requested in light of the following remarks.

### **Section 102(e) Rejection:**

The Examiner rejected claims 1-4, 8-10 and 13-14 under 35 U.S.C. § 102(e) as being anticipated by Conallen (“Building Web Applications with UML”, Second Edition). Applicant respectfully traverses this rejection for at least the following reasons.

In regard to claim 1, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications. On page 3, first paragraph, Conallen states (emphasis added), “this book is about building model-driven Web applications.” The term Web Services is well known in the art, and one of ordinary skill in the art would recognize the difference between the terms Web Services and Web applications. The background section of the instant application provides an extensive discussion of Web Services. Furthermore, Conallen defines Web applications thusly in the paragraph beginning on page 9 and extending onto page 10 (emphasis added):

In its simplest terms, a Web application is a Web system that allows its users to execute business logic with a Web browser...There is a subtle difference between a Web application and a Web site. For the purpose of this book, a Web application is a Web site where user input – navigation through the site and data entry – affects the state of the business: beyond, of course, access logs and hit counters. In essence, a Web application uses a Web site as the front end to a business application.

Conallen does briefly discuss Web Services on pp. 63-68, in a section titled “Web Services” that appears in Chapter 4, titled “Beyond HTTP and HTML,” which begins on page 49. Conallen makes clear the distinction between Web Services and Web applications, for example in the third paragraph on page 63:

The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, it is not a Web application-specific technology. Instead, it uses Web technologies, such as Web

servers and HTTP, to provide a set of services that can be invoked by other programs on the network.

On page 49, second paragraph, Conallen discusses the content and reason for Chapter 4:

With the recent successes of Web applications, more and more architects are choosing this architecture for their next generation of systems. The significant advantages of easy deployment and minimal client configuration are well suited to organizations that maintain a varied array of computer types and models. This increased use of the Web as an architectural platform, however, has stretched the limits of the ability for HTTP and HTML to deliver the functionality required in relatively sophisticated software systems. This chapter discusses the limitations and extensions to these two principal elements of Web applications: HTTP and HTML.

Thus, both Conallen and Applicant's specification are consistent in distinguishing Web Services from Web applications. The term "Web Services" is a well-understood term of art. Most of the teachings of Conallen cited by the Examiner pertain to Web applications, not Web Services as recited in Applicant's claim. **Nowhere does Conallen extend the notions presented in his book to generating integrated Web Service architectures.** Again, Conallen clearly states "this book is about building model-driven Web applications." Conallen's discussion of Web Services is simply an aside "discuss[ing] the limitations and extensions to...two principal elements of Web applications."

**Thus, contrary to the Examiner's assertion, Conallen does not anticipate a system for integrating Web Services with a business system.** The Examiner cites Conallen, page 9, last paragraph, in support of this assertion. However, this citation simply states (emphasis added):

A Web application builds on and extends a Web system to add business functionality.

Again, as noted in the background section of the instant application and in the Conallen reference itself, Web applications are clearly and distinctly different than Web Services.

In further regard to claim 1, Conallen does not anticipate *program instructions executable by the processor to: generate an integrated Web Service architecture*. The Examiner first cites Conallen, page 10, Figure 2-1. Figure 2-1 simply shows a block diagram of a “basic Web system.” Page 10’s text simply includes a portion of a paragraph, cited above, that defines Web applications and thus distinguishes between the notion of Web Services and Web applications, and an introduction to a discussion of HTTP. These citations clearly do not teach the limitations as recited in the claim. These citations do not teach “program instructions executable by the processor to” do anything, much less program instructions executable by a processor to generate an integrated Web Service architecture. Nowhere does Conallen teach *a system for integrating Web Services with a business system, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to: generate an integrated Web Service architecture*. Conallen’s book does not teach any such system comprising a processor and memory comprising program instructions to perform the elements as actually recited in claim 1 when considered as a whole. **In fact, while Conallen’s book is “about building model-driven Web applications,” Conallen does not even teach *a system comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to: generate a Web application architecture*.**

The Examiner then cites pp. 65-66, 64-65, and 66-68 along with page 10, Figure 2-1 as teaching *program instructions executable by the processor to: generate an integrated Web Service architecture*. As noted above, these citations are from Conallen’s discussion of Web Services that is simply an aside “discuss[ing] the limitations and extensions to...two principal elements of Web applications.” These citations simply describe UDDI, SOAP and WSDL, respectively. However, like the above citations, these citations do not teach “program instructions executable by the processor to” do anything, much less program instructions executable by a processor to generate an integrated Web Service architecture. Nor do the two citations in combination teach the limitations as recited in claim 1.

**In further regard to claim 1, Conallen does not anticipate *program instructions executable by the processor to: generate an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system in accordance with one or more integration design patterns*.** The Examiner cites Conallen, page 425, “Master Template Pattern or classes – one example in Figure 6-11, page 115.” Page 425 appears in Appendix D, titled “Master Template Pattern,” which begins on page 423. The first paragraph on page 423, titled “Overview,” states (emphasis added):

The master template mechanism was influenced by the Java Pet Store 1.0.1 example documented in the Java BluePrints. In this mechanism, one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel and providing a single source for updates. This mechanism is most useful for applications that can benefit from an explicitly controlled user interface template.

Page 425 includes further discussion of “screen templates” and “class diagrams.” Appendix D, which discusses a “master template mechanism” in which “one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel” clearly does not describe anything at all like *one or more integration design patterns* used in generating *an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system*. Furthermore, Figure 6-11 on page 115 illustrates a “requirements set”, discussion of which starts on page 114. This Figure and section has nothing whatsoever to do with Appendix D, and, contrary to the Examiner’s assertion, is nowhere in Conallen described as “one example” of a Master Template Pattern. The two citations appear to have little if any relation with one another.

**In further regard to claim 1, Conallen does not anticipate *wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate one or more Use Cases for the integrated Web Service*.** The Examiner cites Conallen, pp. 173-185, 120, 216, 411-414, 101-105, 177-179, 139-141, and 179-183. While the Conallen reference includes several sections that

discuss various “use cases,” nowhere does Conallen disclose one or more Use Cases for an integrated Web Service. As previously noted, Conallen is generally directed at building Web applications (“this book is about building model-driven Web applications”), not Web Services. The citations provided by the Examiner disclose various Web application use cases. On page 173, one of the pages cited by the Examiner, at the beginning of a section titled “Use Cases”, Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications.

Furthermore, while Conallen does disclose various Use Cases (again, Web application-specific use cases), nowhere does Conallen disclose *program instructions executable by a processor to generate one or more Use Cases for an integrated Web Service*. In fact, Conallen does not even disclose *program instructions executable by a processor to generate one or more Use Cases* in reference to the Web application-specific Use Cases Conallen does discuss.

**In further regard to claim 1, Conallen does not anticipate *wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service***. The Examiner asserts “as per above” in reference to this limitation. Applicant’s above arguments make it clear that Conallen is not even directed at generating architectures for integrated Web Services. Furthermore, Conallen does not even disclose *program instructions executable by a processor to generate a high-level architecture* of any type. Conallen nowhere discloses any such computer-executable program instructions.

**In further regard to claim 1, Conallen does not anticipate *wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the***

***integrated Web Service.*** The Examiner cites Conallen, page 438. This citation is from Appendix E, “Glossary Application,” that begins on page 429. The second paragraph on page 429 states (emphasis added):

The overall goal and vision for this application is to demonstrate, in the context of a simple and functional application, a technique for modeling Web applications with UML.

This Appendix is clearly directed at a technique for modeling Web applications, and an example thereof, and not at Web Services, and thus none of the examples from this Appendix, including page 438, illustrate anything like *a high-level architecture[for an integrated Web Service] that identifies two or more entities of the integrated Web Service*.

**In further regard to claim 1, Conallen does not anticipate *wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to: generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities*.** The Examiner cites page 177, “relationship.” This citation appears in a section titled “The Use Case Model” that begins on page 176, and specifically discusses relationships between “actors” and Use Cases (page 177, first paragraph) and relationships between Use Cases (page 177, second paragraph). This citation clearly describes nothing like what is actually recited in claim 1 (the relationship among two or more entities of an integrated Web Service).

**In further regard to claim 1, Conallen does not anticipate *wherein, to generate an integrated Web Service architecture, the program instructions are further executable by the processor to generate a logical architecture for the integrated Web Service according to the Use Cases, wherein the logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers*.** The Examiner cites Conallen, pages 237-242. This citation is in a section titled “Web

Application Extension [WAE] for UML” that begins on page 236. Again, this section is directed at Web applications, not Web Services. The subsection beginning on page 237, cited by the Examiner, is titled “Logical View.” This subsection, like the rest of the section, is clearly directed at Web applications, not Web Services.

Furthermore, Conallen, in this section or elsewhere, does not even disclose *program instructions executable by a processor to generate a logical architecture* of any type.

Applicant reminds the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention must** be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As shown by the Applicant’s above arguments, nowhere does the Conallen reference disclose “each and every element of the claimed invention” (claim 1 of the instant application) as arranged in the claim. Furthermore, even if Conallen did disclose one or more of the above elements, nowhere does Conallen disclose the above elements arranged as in claim 1. Moreover, the Examiner has improperly cited portions of Conallen from various chapters, sections, and appendices, some of which do not appear to be directly related, in an attempt to assert that Conallen discloses what is recited in claim 1. For at least the reasons given above, Conallen clearly does not anticipate Applicant’s claim 1.

### **Section 103(a) Rejection:**

The Examiner rejected claims 5-7 and 11-12 under 35 U.S.C. § 103(a) as being unpatentable over Web in view of Taylor (“Object-Oriented Information Systems Planning and Implementation”). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time.



The Examiner rejected claims 15 and 16 under 5 U.S.C. § 103(a) as being unpatentable over Web in view of Smith (“The Hazards of Closed-Process Development”). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time.

The Examiner rejected claim 17 as being unpatentable over Web as applied to claim 1 above, and further in view of Boucher, et al. (“Essential Guide to Object Monitors”). Since the rejection of the independent claim has been shown to be unsupported by the cited references, a further discussion of this rejection is not necessary at this time.

Applicant also asserts that the rejection of numerous ones of the dependent claims is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

## **CONCLUSION**

Applicant submits the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-66303/RCK.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Applicant(s)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: February 26, 2008